Lucky Beulla Muhoza

CS338: Computer security

Prof. Jeff Ondich

October 7th, 2024

# Cryptographic scenarios

## Assumptions

Suppose Alice, Bob, Eve, Mal, and all their friends and enemies have access to the following.

- The symmetric encryption algorithm AES. Use the function name AES to denote this: AES(K, M) is the message M encrypted using the key K. To denote decryption of the ciphertext C, use AES_D(K, C). Assume everyone has agreed on a suitable block cipher mode (e.g., CBC).
  Don't forget that before Alice and Bob can use this algorithm, they have to agree on a key K, which is not automatically provided to them.
- A Diffie-Hellman key exchange procedure. If you want to use this, just say "Alice and Bob use Diffie-Hellman to agree on a shared secret from which they derive an AES key K" or something like that.
- The cryptographic hash function SHA-256. Represent the hash of a message M by H(M).
- Public/secret key pairs (P, S) for everybody (we'll use "secret" and "private" interchangeably when talking about these keys). Denote Alice's key pair as (P_A, S_A), Bob's as (P_B, S_B), etc.
  For encryption and decryption with the public and secret keys, use the function E. For example, if M is a small enough message to be in the domain of E, then Bob can send an encryption of M to Alice by sending her the ciphertext C = E(P_A, M). Then Alice can compute E(S_A, C) = E(S_A, E(P_A, M)) = M to retrieve Bob's message.
  Keep in mind that in practice, public key encryption is generally used for short messages (e.g., to encrypt a hash function digest), mainly because it is much slower than symmetric encryption.
  Unless otherwise instructed, you may assume that everybody has a correct copy of everybody else's public key, and that they have all kept their private keys secret. This assumption is a big one—exchanging public keys safely is a hard problem, as we have started to see in our initial investigations of certificates.

## Simple communication scenarios

For each of the scenarios below, describe as concisely as you can how you would use the tools listed above to achieve the goals described in the scenario. Then, briefly explain why your plan achieves those goals.

Make your plans as simple as possible given the goals of the scenario. You might be able to come up with a single plan that handles all the scenarios, but that's not what I'm after. I want you to understand the properties of Diffie Hellman, symmetric encryption, public-key encryption, cryptographic hashes, digital signatures, etc. By responding to each scenario with the simplest plan using the available tools, you'll demonstrate that understanding.

We'll use Eve to refer to any eavesdropper, and Mal to refer to any person attempting an adversary-in-the-middle (AITM) attack.

1. Alice wants to send Bob a long message, and she doesn't want Eve to be able to read it. Assume for this scenario that AITM is impossible.

   *For Alice to ensure the **confidentiality** of her message to Bob, she needs to **encrypt** her message.*

- *Alice and Bob use Diffie-Hellman to agree on a shared secret from which they derive an AES key K.*
- *Alice then encrypts her message M using AES(K, M) and sends Bob the resulting ciphertext C.*
- *Bob uses AES_D(K, C) to decrypt the ciphertext C, and hence retrieves his message M.*

   *This plan ensures that only Alice and Bob, who share the key K, can encrypt and decrypt the message and Eve cannot read the message since she does not have the shared secret key.*

2. Alice wants to send Bob a long message. She doesn't want Mal to be able to modify the message without Bob detecting the change.

   *To ensure the **integrity** of her message, Alice has to use a **cryptographic hash function**.*

- *Alice and Bob use Diffie-Hellman to agree on a shared secret from which they derive an AES key K.*
- *Alice computes a hash of her message M using SHA-256 to obtain H(M).*
- *Alice then concatenates H(M) to M, forming M’ = M || H(M), and then encrypts M’ using AES(K, M’) to get C.*
- *Alice sends C to Bob who decrypts the ciphertext using AES_D(K, C) to retrieve M’. He then splits M’ to H(M) get M.*

*To verify integrity, Bob computes H(M) from the retrieved M independently and compares it with Alice's appended H(M). If they match, Bob can be confident that the message has not been modified.*

3. Alice wants to send Bob a long message (in this case, it's a signed contract between AliceCom and BobCom), she doesn't want Eve to be able to read it, and she wants Bob to have confidence that it was Alice who sent the message. Assume for this scenario that AITM is impossible.

   *Alice has to combine encryption with a **digital signature or a MAC** to assure Bob it was her who sent it (**authenticity**).*

1. *Alice and Bob use Diffie-Hellman to agree on a shared secret from which they derive an AES key K.*
2. *Alice computes H(M) using a hash function and then signs this hash with her private key, $S_A$, creating a signature $Sig=E(S_A, H(M))$.*
3. *Alice combines both M and Sig, creating M' = M || Sig. She then encrypts M' which she sends to Bob.*
4. *Bob separates M' to retrieve M and Sig. He decrypts Sig using Alice's public key $P_A$ to obtain H(M)  and compares the hashed result to the result of hashing Alice's message  M. If the verification is successful, Bob can be sure the message came from Alice and was not altered.*

   *By combining encryption (AES) with a digital signature, Alice not only keeps the message confidential but also provides proof of authenticity. The signature ensures that Bob can confirm that only Alice could have sent the message.*

## Questions about breaking security

In the following, the symbol || represents concatenation. For example, if X and Y are 4-byte integers, then X || Y is the 8-byte quantity consisting of X followed by Y. Or "dog" || "house" is the string "doghouse". We won't worry about issues like byte-order or encoding schemes for this assignment.

4. Consider a scenario where Alice and Bob have been in contract negotiations and sharing documents electronically along the way. Suppose Bob sues Alice for breach of contract and presents as evidence the digitally signed contract (C || Sig) and Alice's public key P_A. Here, C contains some indication that Alice has agreed to the contract—e.g., if C is a PDF file containing an image of Alice's handwritten signature. Sig, on the other hand, is a digital signature, as described at 9:23 or so of the [Cryptographic Hash Functions video](). Suppose Alice says in court "C is not the contract I sent to Bob". (This is known as *repudiation* in cryptographic vocabulary.) Alice will now need to explain to the court

what she believes happened that enabled Bob to end up with an erroneous contract. List at least three things Alice could claim happened. For each of Alice's claims, state briefly how plausible you would find the claim if you were the judge. (Assume that you, the judge, studied cryptography in college.)

- *An adversary-in-the-middle (AITM) attack might have occurred where a malefactor Mal might have substituted Alice's public key P_A with a different one. This would happen if their public key exchange happened over an untrustworthy third party certificate authority.*
- *Bob himself might have altered the contents of the contact C without changing the signature Sig by performing collision attacks on the used hash function to create two different contracts with the same hash.*
- *Alice's Sig might have been obtained from a previous interaction between the two parties and reused to falsely authenticate a new contract C.*

5. For this scenario, suppose the assumption that everybody has everybody else's correct public keys is no longer true. Instead, suppose we now have a certificate authority CA, and that everybody has the correct P_CA (i.e. the certificate authority's key). Suppose further that Bob sent his public key P_B to CA, and that CA then delivered to Bob this certificate:
   In terms of P_CA, S_CA, H, E, etc., what would Sig_CA consist of? That is, show the formula CA would use to compute Sig_CA.

   → *Sig_CA = E (S_CA, H (data)) where data contains Bob's public key P_B sent to CA and other relevant metadata. If CA sends Bob the certificate Cert_B, anyone can use the certificate authority's key P_CA to verify Bob's Sig_CA.*

6. Bob now has the certificate Cert_B from the previous question. During a communication, Bob sends Alice Cert_B. Is that enough for Alice to believe she's talking to Bob? (Hint: no.) What could Alice and Bob do to convince Alice that Bob has the S_B that goes with the P_B in Cert_B?

   *Alice can validate Bob's certificate Cert_B by computing X and Y*

   *Where:*

- *X = H (data)*
- *Y = E (P_CA, Sig_CA)*

*Alice will be convinced that Bob has S_B that goes with P_B in Cert_B if the computed X == Y.*
*And if Cert_B is legit, Y = E (P_CA, Sig_CA)  = E (P_CA, E (S_CA, H (data))) = H (data) = X*

7. Finally, list at least two ways the certificate-based trust system from the previous two questions could be subverted, allowing Mal to convince Alice that Mal is Bob.

   - *Mal can conduct an adversary-in-the-middle (AITM) attack between Alice and Bob's communication, issuing a forged certificate to Alice. If Alice does not independently verify Cert_B legitimacy, she might be tricked into believing it is Bob's.*
   - *Mal could request a certificate from CA using Bob's identity. If Mal somehow successfully convinces CA that she is the legit owner of Bob's public key P_B, then a certificate can be issued to Mal in the name of Bob hence Alice, trusting CA, is manipulated to also believe Mal is Bob.*